



INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS								
V Semester: CSE								
Course Code	Category	Hours/Week			Credits	Maximum Marks		
ACSD38	Elective	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil			Total Classes: 48			
Prerequisite: Object Oriented Programming								

I. COURSE OVERVIEW:

This course focuses on the principles, concepts, and techniques used in designing and architecting software systems. It emphasizes understanding the critical decisions and trade-offs involved in designing robust, scalable, maintainable, and high-performance software architectures. Students will explore various architectural patterns, design principles, and best practices, along with real-world case studies to understand how architectural decisions impact software development and lifecycle.

II. COURSE OBJECTIVES:

The students will try to learn:

- I The foundational concepts of software architecture, its significance, and various architectural styles and patterns used in real-world software systems.
- II The design principles (such as SOLID and GRASP) and design patterns (Creational, Structural, Behavioral) to create effective and scalable software solutions.
- III The ability to identify suitable architectural and design patterns, recognize anti-patterns, and apply refactoring techniques to improve software quality through practical implementation.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

- CO1 Outline process models, approaches and techniques the architecture, creating Understand it and moving from one to any, different structural patterns
- CO2 Evaluate professional ethics in types of the various applications of Apply Design creational and structural patterns.
- CO3 Elaborate the life cycle of the Learn about behavioral patterns
- CO4 Analyze evaluation of organization and core metrics for Establishing infrastructure Understand Calculations
- CO5 Apply model based architectural concepts for routing algorithms for different network environment.
- CO6 Determine case study on future software project management practices in Do Apply cease studyir utilizing architectural structures

IV. COURSE CONTENT:

MODULE - I: INTRODUCTION TO SOFTWARE ARCHITECTURE (10)

Overview of software architecture definition and importance role of a software architect architectural patterns vs. design patterns, architectural styles layered architecture, client-server architecture, event-driven architecture, microservices architecture case studies in software architecture real-world examples analysis and discussion.

MODULE - II: ARCHITECTURAL PATTERNS (09)

Introduction to architectural patterns, definition and purpose overview of common patterns, layered pattern, concept and structure, use cases and examples, client-server and microservices patterns, details of each pattern advantages and disadvantages, event-driven and service-oriented architectures, key concepts and implementation examples and best practices.

MODULE - III: DESIGN PRINCIPLES AND PRACTICES (10)

Solid principles, single responsibility principle, open/closed principle IISKOV substitution principle interface segregation principle, dependency inversion principle, grasp principles general responsibility assignment software patterns.

Information expert, creator, controller low coupling, high cohesion, design documentation diagrams design documentation best practices.

MODULE – IV: DESIGN PATTERNS (09)

Introduction to Design Patterns, Definition and importance Types of design patterns (Creational, Structural, Behavioral) Creational Patterns, Singleton, Factory Method, Abstract Factory, Builder, Prototype Structural Patterns, Adapter, Composite, Proxy, Flyweight, Facade, Bridge, Decorator Behavioral Patterns, Strategy, Observer, Command, Mediator, Chain of Responsibility, Template Method, State.

MODULE – V: APPLYING ARCHITECTURE AND DESIGN PATTERNS (10)

Pattern Application and Anti-Patterns, identifying suitable patterns for problems, recognizing and avoiding anti-patterns, Refactoring and Design Improvements, Code smells, Techniques for refactoring, Capstone Project Design, and implementing a small software system using the principles and patterns learned.

V. TEXT BOOKS:

1. Len Bass et al., *Software Architecture in Practice*, Addison-Wesley, 4th Edition., 2021.
2. Erich Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1st Edition, 1994.

VI. REFERENCE BOOKS:

1. Robert C. Martin, “*Clean Architecture*”, Pearson, 1st Edition, 2017.
2. R. N. Taylor et al., “*Software Architecture: Foundations, Theory, and Practice*”, Wiley, 1st Edition, 2009.
3. Frank Buchmann et al., “*Pattern-Oriented Software Architecture: A System of Patterns*”, Wiley, 1st Edition., 1996.

VII. ELECTRONICS RESOURCES:

1. <https://c4model.com/>
2. <https://www.coursera.org/learn/software-design-architecture>
3. <https://www.coursera.org/specializations/cloud-architecture>

VIII. MATERIAL ONLINE:

1. Course template
2. Tutorial question bank
3. Tech-talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)